DEBUGGING APPARATUS AND METHOD

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a debugging apparatus and method, and more particularly, to a debugging apparatus and method that is capable of recognizing a data conversion ~~state~~ in a specific memory location by observing a change and flow of ~~a~~ specific data ~~of a~~ memory location and performing a debugging~~, and its method~~ process.

2. Description of the Background Art

In general, a debugging apparatus~~,~~ for detecting an error generated ~~to~~ in a program input~~ted~~ to a microprocessor~~,~~ includes a host computer and a microprocessor unit (MPU).

In the debugging apparatus, when the host computer selects a specific address of a program memory in the MPU as a break point, the MPU monitors the processor while the processor is being operated~~,~~ and when the specific address of the program memory accessed is identical to the memory address selected as the break point, the MPU recognizes the selected memory address as a break point and discontinues the operation of the processor and the host computer observes ~~a~~

the flow of a the program and performs a debugging ~~on the program~~ process.

Figure 1 is a schematic block diagram of the debugging apparatus in accordance with a the conventional art.

As shown in Figure 1, the debugging apparatus includes: a host computer 1 for selecting a specific address of a memory as a break point and performing a debugging process; a debugger controller 2 for receiving a control command from the host computer 1 and outputting a break enable signal and a break point address; a processor ~~core~~ 4 ~~being operated~~ operating upon receiving a control signal from the debugger controller 2; a program memory 5 for storing a program of the processor ~~core~~ 4; a data memory 6 for storing a data of the processor ~~core~~ 4; and a break point sensing unit 3 for receiving the break enable signal and the break point address from the debugger controller 2, observing an address of the program memory 5 being executed in the processor ~~core~~ 4, and recognizing ~~an~~ the address as a ~~debugger~~ break point and outputting a break signal to the debugger controller 2 ~~if the address is sensed to be identical to the inputted break point address~~.

The operation of the debugging apparatus constructed as described above will now be explained.

When the processor is switched to a debugging mode, the host computer 1 outputs a processor ~~core~~ stop signal and a break point address to the debugger controller 2~~,~~ for debugging.

The debugger controller 2 outputs a stop signal to the processor ~~core~~ 4 to suspend the processor ~~core~~ 4~~,~~ operation and outputs a break point address and a

break enable signal to the break point sensing unit 3.

When the break point sensing unit 3 stores a program address ~~that~~ in the program memory at which the processor ~~core~~ 4 ~~wants~~ is to suspend operation ~~in the program memory~~, ~~that is, stores~~ or a a break point address, the host computer 1 operates the processor ~~core~~ 4 in the order of programs stored in the program memory 5.

While the processor ~~core~~ 4 is operated ~~in the program order~~, the break point sensing unit 3 observes a program address output~~ted~~ to the processor ~~core~~ 4.

Subsequently, when the break point sensing unit 3 detects an accessed address of the ~~same~~ program memory 5 as that of the stored program address, ~~it outputs~~ a break signal is output to the debugger controller 2.

The debugger controller 2 suspends the operation of the processor ~~core~~ 4 according to the break signal is ~~inputted~~ received from the break point sensing unit 3 and shifts ~~a~~ debugging control ~~right~~ to the host computer 1~~,~~ so that the host computer 1 may perform a debugging operation.

However, in the debugging method, since an operation is determined by an address of a specific program memory among the sequential programs, it is not possible to recognize a data flow according to a data memory. Thus, in case where an error occurs by a data, much time and expense is taken for debugging.

In addition, ~~in case of~~ when creating a program, since the state of a data is ~~hardly~~ not recognized, an error ~~occurs~~ may occur in ~~a data memory~~ the allocation ~~amount~~ of data memory. In addition, ~~in case of~~ if the wrong data is read~~ing out a~~

~~wrong data~~ in a calculating process, an erroneous result ~~value~~ is output~~ted only to~~ and may cause a system malfunction ~~to a system~~.

SUMMARY OF THE INVENTION

Therefore, an object of the present invention is to provide a debugging apparatus and method that is capable of saving a time and an expense in performing a debugging operation by recognizing a data transition state in a specific data memory by observing a change ~~state and flow~~ of ~~an~~ a data address and a change of data ~~of~~ at the specific ~~data memory~~ address in ~~a~~ data memory.

To achieve these and other advantages and in accordance with the purpose of the present invention, as embodied and broadly described herein, there is provided a debugging apparatus including: a processor ~~core~~ operated by a program stored in a program memory to read ~~a~~ data stored in ~~a~~ data memory or write ~~a~~ data to data memory; a debugger controller for performing a debugging process on the processor ~~core~~ upon receipt of a command from a host computer and outputting a data break point address; and a memory break controller for observing an address of a data memory ~~used~~ accessed by the processor ~~core~~, recognizing an accessed address as a break point address ~~to output~~ and outputting a break signal to the debugger controller and the processor ~~core~~ to suspend the operation of the processor ~~core~~, when the address ~~is sensed to be~~ accessed is identical to the address to be observed, and transmitting ~~a~~ ~~corresponding~~ the address and corresponding data to the host computer through

the debugger controller.

To achieve the above objects, there is further provided a debugging method including the steps of: outputting an address of a data memory to be observed, ~~that is~~ or a break point address, and a break enable signal, when a processor is switched to a debugging mode; storing the output~~ted~~ break point address, and operating the processor in a general operation state; comparing the stored break point address and the address of the data memory currently ~~used~~ accessed by the processor ~~core, while the process is being operated~~; outputting a break signal to suspend operation of the processor ~~core,~~ if the address of the data memory currently ~~used~~ accessed by the processor ~~core~~ and the stored break point address are identical ~~to each other~~; and suspending operation of the processor ~~core~~ by the output~~ted~~ break signal and switching the processor to a debugging mode to debug the program.

The foregoing and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and together with the description serve to explain the principles of the invention.

In the drawings:

Figure 1 is a schematic block diagram showing a debugging apparatus in accordance with a conventional art;

Figure 2 is a schematic block diagram showing a debugging apparatus in accordance with a preferred embodiment of the present invention;

Figure 3 is a detailed block diagram showing a memory break controller of Figure 2 in accordance with the preferred embodiment of the present invention;

Figure 4 is a schematic block diagram showing the structure of a memory break control register of Figure 3 in accordance with the preferred embodiment of the present invention; and

Figure 5 is a flow chart of a debugging method in accordance with the preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings.

Figure 2 is a schematic block diagram showing a debugging apparatus in accordance with a preferred embodiment of the present invention.

As shown in Figure 2, a debugging apparatus of the present invention includes a host computer 10 for assigning a break point address and controlling a debugging operation; a debugger controller 20 for outputting a control signal to control a processor ~~outputted from the host computer 10~~, a break point address

and a break enable signal; a program memory 60 for storing a program to operate a processor ~~core~~ 40 ~~(to be described)~~; a processor ~~core~~ 40 ~~being~~ operated ~~by~~ according to the control signal output~~ted~~ from the debugger controller 20 and the program stored in the program memory ~~60, and outputting a data generated accordingly~~; a data memory 70 for storing ~~a~~ data output~~ted~~ from the processor ~~core~~ 40; a break point sensing unit 30 for observing an address of the program memory 60 ~~used~~ accessed by the processor ~~core~~ 40, ~~and~~ recognizing an address as a debugger break point and transmitting a break signal to the debugger controller 20~~,~~ if the accessed address is ~~sensed to be~~ identical to the break point address output~~ted~~ from the debugger controller 20; and a memory break controller 50 for observing an address and ~~a~~ data of the data memory 70 ~~used~~ accessed by the processor ~~core~~ 40, ~~and~~ transmitting the ~~sensed~~ currently accessed address and ~~it's~~ the corresponding data to the host computer 10 though the debugger controller 20 and activating an operation of the processor ~~core~~ 40 to transmit the address and data of the data memory ~~used in~~ accessed by the processor ~~core~~ 40 to the host computer 10 through the debugger controller 20 until a break signal is output~~ted~~ again~~,~~ when the same address as the break address output~~ted~~ from the debugger controller 20 is ~~sensed~~ identified.

The host computer 10 recognizes a data flow and change according to the address and the data output~~ted~~ from the memory break controller 50.

The construction of the memory break controller of the debugging apparatus will now be described with reference to Figure 3.

Figure 3 is a detailed block diagram showing a memory break controller of

Figure 2 in accordance with the preferred embodiment of the present invention.

As shown in Figure 3, the memory break controller 50 includes a memory break control register 51 being activated by a control signal and a break point address signal output~~ted~~ from the debugger controller 20 and observing a data change of the processor ~~core~~ 40; an address register (AR) 52 for storing the break point address output~~ted~~ from the memory break control register 51; an address comparator (AC) 53 for comparing the address currently output~~ted~~ from the data memory 70 and the break point address stored in the address register 52; a data register (DR) 54 for storing ~~a~~ data ~~of~~ stored at the break point address ~~stored in the address register 52~~; and a data comparator (DC) 55 for comparing the data ~~of~~ stored at the break point address of ~~outputted from~~ the data memory 70 and the data value of the break point address stored in the data register 54.

The construction of the memory break control register 51 will now be described with reference to Figure 4.

Figure 4 is a schematic block diagram showing the structure of a memory break control register of Figure 3 in accordance with the preferred embodiment of the present invention.

As shown in Figure 4, the memory break control register 51 includes: a memory break enable flag (MBEF) 51-1 for activating the memory break controller 50; a data check flag (DCF) 51-2 for sensing an address of the data memory which is identical to the break point address stored in the address register 52 and being enabled when the data of the corresponding address is output~~ted~~; and an address trace check flag (ACF) 51-3 assigning an initial break point address,

being enabled when ~~a~~ the content of the break point address is read during the processing procedure, and outputting the ~~content of the~~ data and the addresses of every memory location read from or written to ~~from or into~~ by the processor until the content of the break address is updated.

The operation of the debugging operation constructed as described above will now be explained with reference to Figure 5.

Figure 5 is a flow chart of a debugging method in accordance with the preferred embodiment of the present invention.

As shown in Figure 5, while the processor in the host computer 10 is being ~~operation, when~~ operated and ~~it~~ is switched to a debugger mode, the host computer 10 disables the debugger mode, so that the host computer sets a start position of a debugger mode program as a start position of the processor and outputs an initialization control signal and an address of the data memory to be observed, ~~that is,~~ or a break point address, to the debugger controller 20.

According to the control signal and the break point address output~~ted~~ from the host computer 10, the debugger controller 20 outputs the control signal, the break point address and the break enable signal to the memory break controller 50 and the break point sensing unit 30.

The break point sensing unit 30 receives and stores the break enable signal and the break point address.

According to the ~~inputted~~ received break enable signal, the memory break control register 51 of the memory break controller 50 enables a memory break enable flag (MBEF) as '1' and disables a data check flag (DCF) and an address

~~track~~ trace check flag (ACF) as '0' (step S10), thereby initializing the processor.

The address register 52 of the memory break controller 50 stores the inputted break point address (step S11).

After completing the process, the host computer 10 operates the processor ~~core~~ 40 according to programs stored in the program memory 60.

When the processor ~~core~~ 40 is operated according to the program sequentially stored in the program memory, the break point sensing unit 30 monitors whether the break point address stored in the processor ~~core~~ 40 and the address of the program memory 60 ~~used~~ accessed by the processor ~~core~~ 40 are identical ~~to each other~~.

The address comparator 53 (AC) of the memory break controller 50 compares the address of the data memory 70 used by the processor ~~core~~ 40 and the break point address (step S12).

~~Upon comparison, if~~ If the address of the data memory 70 ~~read from~~ accessed by the processor ~~core~~ 40 and the break point address stored in the address register 52 are identical ~~to each other, that is, if~~ such that the address comparator 53 is enabled, the address comparator 53 outputs an accord signal to the debugger controller 20.

The debugger controller 20 ~~judges~~ determines whether the processor ~~core~~ 40 is reading ~~a~~ data ~~for~~ from the ~~corresponding~~ address of the data memory 70 or writing ~~a~~ data to the address of the data memory 70 (step S14).

~~According to judgement result, in case that~~ If the processor ~~core~~ 40 ~~writes a~~ is writing data ~~in~~ to the data memory 70, the debugger controller 20 enables a

data check flag (DCF) ~~(~~51-2~~)~~ of the memory break control register 51 (step S18) and outputs a break signal to the processor for suspending execution of the program of the processor ~~core~~ 40 ~~to the the processor core 40~~ (step S20)~~,~~ ~~so as to discontinue~~ such that the operation of the processor ~~core~~ is discontinued.

~~Meanwhile, in case that~~ If the processor ~~core~~ 40 reads a data ~~of an~~ from the address of the ~~corresponding~~ data memory 70, the debugger controller 20 enables an address trace check flag (ACF) 51-3 and a data check flag (~~CDF~~ DCF) 51-2 of the memory break control register 51 (step S15).

The data ~~of~~ at a specific address of the data memory (70) read by the processor ~~core~~ 40 ~~has~~ may have an arithmetic and logical ~~operation~~ relation ~~with a~~ to the data ~~value of~~ at a ~~difference~~ different address or a correlation with ~~a~~ another specific address of the data memory 70.

Accordingly, ~~in case if that a~~ if data of a specific address is read, when the address trace check flag (ACF) 51-3 and the data check flag (DCF) 51-2 are enabled, the address comparator 53 compares the specific address and the break point address (step S16).

When the break point address stored in the address register 52 and the accessed address of the data memory 70 are identical ~~to each other~~, the address comparator 53 transmits an accord signal to the debugger controller 30.

If, however, the break point address and the address of the data memory are not identical ~~to each other~~ such~~,~~ that ~~is, if~~ the value ~~(~~AC~~)~~ of the address comparator (AC) 53 is not '1', since the break point address and the address to be observed are different arithmetically and logically, the address comparator 53 and

the data comparator 55 transmit the ~~sensed~~ address and ~~data of the~~ corresponding data ~~memory~~ to the host computer 10 through the debugger controller 20~~,~~ for a debugging operation.(step S17).

~~Meanwhile, when~~ If the address comparator (AC) 53 is enabled, it ~~means~~ indicates that the previously read address to be observed is ~~used,~~ accessed and the debugger controller 20 determines whether the processor ~~core~~ 40 is reading a data ~~of~~ from the address ~~to be observed of the data memory 70~~ or a writing data ~~value is writing into~~ to the address ~~to be observed~~ (step S19).

~~In case that a~~ If data is being read from ~~of~~ the ~~address to be observed of~~ ~~the~~ data memory 70 address to be observed ~~is being read~~, since it is the previously read address ~~to be observed~~, ~~it returns to the step in which~~ the address comparator 53 of the memory break controller 50 compares the next read address and the break point address ~~and~~ to determine~~s~~ whether the read address is identical to the break point address stored in the address register 52 (step S16).

~~Meanwhile, in case that~~ If a data value is being written ~~in a~~ to the address to be observed, since ~~it means that~~ a result value according to an arithmetic operation is being written in the previously read address ~~to be observed~~, the memory break controller 50 outputs a break signal to the debugger controller 20 and the processor ~~core~~ 40 (step S20)~~,~~ in order to discontinue the operation of the processor ~~core~~ 40.

Upon receiving the break signal, the debugger controller 20 outputs a signal allowing the host computer 10 to start a debugging operation. The host computer 10 performs a debugging operation according to the output~~ted~~ signal.

Meanwhile, if the address to be observed and the address ~~comparison result~~ of the data memory currently ~~used~~ accessed by the processor ~~core~~ 40 are different ~~to each other~~ (step S12), the memory break controller 50 outputs the address and data of the data memory 70 location currently ~~read~~ accessed by the processor ~~core~~ 40 to the host computer 10.

Subsequently, the address comparator 53 ~~repeats from the step for comparing~~ compares the address of the next data memory 70 ~~used~~ location accessed by the processor ~~core~~ 40.

The operation of the debugging apparatus as described above will now be explained in detail by ~~taking~~ considering the following program as an example.

A0 = a ---------- (1)

A1 = b ---------- (2)

A2 = c ---------- (3)

A2 = A2 + (A1 * A2) ---------- (4)

wherein A0, A1 and A2 are addresses of the data memory. The program stores initial values a, b and c in, respectively, memory locations A0, A1 and A2 in steps 1-3. The program then updates the value stored in memory location A2, in step 4, based on the current value of memory location A2 and the initial value of A1.

In the program constructed as described above, the host computer 10 designates an address to be observed as A2 and executes the program in a debug mode.

The address comparator 53 performs ~~executes~~ a debugging process by ~~while~~ comparing A2, the address to be observed, and the currently ~~used~~ accessed

address~~,~~. When ~~and when~~ the address comparator 53 reads a data value 'c' corresponding to A2, it senses A2 as ~~an~~ the address to be observed.

When the address comparator 53 ~~senses~~ identifies A2 as the address to be observed, the memory break control register 51 outputs a break signal to the processor ~~core~~ 40 in order to discontinue ~~the~~ program operation, and determines whether A2 of formula (3) is ~~for~~ being read~~ing~~ from or ~~writing~~ written to.

If ~~Upon determining, if~~ A2 is ~~for~~ being read~~ing~~ from, the memory break control register 51 transmits the A2 address and the data value 'c' to the host computer 10.

If ~~In case of~~ executing continuously, the memory break controller 50 reads ~~A2, so that~~ addresses of every memory ~~related~~ location accessed until the next ~~A2~~ value is written to A2 and the~~ir~~ contents of the memory location are output~~ted~~. That is, the address comparator 53 continuously executes the processor, stops ~~at~~ when A2 ~~of the~~ is accessed in formula (4), and compares A0, A1 and A2 with ~~A2,~~ the address to be observed.

After the address comparator 53 ~~first~~ compares A0 and A1 with the address to be observed ~~A2~~, ~~since~~ it is determined that A0 and A1 are not identical to the address to be observed~~,~~ and the address comparator 53 transmits a corresponding address and data value to the host computer 10. ~~and~~ The address comparator then compares A2 of the right side of ~~the~~ formula (4) with the address to be observed.

~~Wince~~ Since A2 is identical to the address to be observed, the memory break control register 51 determines ~~whether it~~ if A2 is read~~ing~~ from or ~~writing~~

written to.

~~In this respect,~~ Address A2 as presented in ~~the~~ formula (3) is ~~an~~ the address which ~~has been~~ was read from in the previous process, ~~it goes~~ so the address comparator moves to the next address.

Subsequently, after ~~the right side of~~ A2 on the right side of ~~the~~ formula (4) is compared, A2 on the left side of formula (4) is compared. ~~In this respect, after A2 is sensed, it~~ It is determined whether ~~it~~ A2 is being read~~ing~~ from or ~~writing~~ written to.

Since A2 on the left side of ~~the~~ formula (4) is ~~for writing~~ being written to, the memory break controller 50 outputs a break signal to the debugger controller 20 and the processor ~~core~~ 40 to suspend the operation of the processor ~~core~~ 40.

The debugger controller 20 outputs a debugging mode switch signal to the host computer 10 ~~by~~ in response to the ~~inputted~~ break signal.

Upon receiving the debugging mode switch signal from the debugger controller 20, the host computer 10 ~~operates~~ updates the data value stored in A2 ~~by~~ according to the previously transmitted address and data ~~and updates A2 of the data memory to the operated value~~.

~~In case that~~ If a range of data ~~constructed with arrangement~~ addresses is stored in the address register 52 ~~is observed~~, the host computer 10 designates a break point address as an uppermost address and a lowermost address ~~of an arrangement of~~ corresponding to the range of addresses in register 52 ~~of the memory break controller 50~~.

The address register 52 stores the uppermost address and lowermost

address of the ~~arrangement of the~~ address range in data register 54~~,~~ and sets a space for storing ~~a~~ data stored in ~~of~~ the ~~arrangement of the~~ range of data addresses stored in data register 54.

The address comparator 53 compares the range of ~~the arrangement~~ addresses stored in the address register ~~and~~ with the ~~break point~~ currently accessed address and ~~senses~~ identifies an address ~~of the data memory~~ corresponding to the range of ~~the arrangement~~ addresses. The ~~following~~ operation is performed in the same manner as the case of observing one data address.

As so far described, the debugging apparatus and method of the present invention has the following advantages.

~~That is, for example, in~~ In a debugging operation, since an address and a data of a specific data memory are monitored to recognize a data flow and change of the specific address, an error ~~that~~ caused by an erroneous calculation ~~is inputted a~~ during processing or erroneously assigning a data memory ~~is erroneously assigned~~ location is quickly sensed. Thus, a time and an expense for a debugging operation can be much saved.

In addition, by adding ~~a~~ the data debugging method of the present invention to ~~the~~ a conventional program debugging method, ~~a~~ the program development environment is ~~set~~ similar to an environment ~~that~~ in which the processor is ~~substantially~~ operated. Thus, ~~a~~ program development time and ~~an~~ expense ~~for developing a program can~~ may be ~~also~~ saved.

As the present invention may be embodied in several forms without departing from the spirit or essential characteristics thereof, it should also be

understood that the above-described embodiments are not limited by any of the details of the foregoing description, unless otherwise specified, but rather should be construed broadly within its spirit and scope as defined in the appended claims, and therefore all changes and modifications that fall within the meets and bounds of the claims, or equivalence of such meets and bounds are therefore intended to be embraced by the appended claims.

What is claimed is:

1. A debugging apparatus comprising:

a processor core operated by a program stored in a program memory to read a data stored in a data memory or write a data;

a debugger controller for performing a debugging on the processor core upon receipt of a command from a host computer and outputting a data break point address; and

a memory break controller for observing an address of a data memory used by the processor core, recognizing an address as a break point address to output a break signal to the debugger controller and the processor core to suspend the operation of the processor core, when the address is sensed to be identical, and transmitting a corresponding address and data to the host computer through the debugger controller.

2. The apparatus of claim 1, wherein the data memory stores a data value outputted from the processor core.

3. The apparatus of claim 1, wherein the memory break controller transmits an address and a data recognized as a break point, activates an operation of the processor core, and outputs the used address and the data of the data memory to the host computer through the debugger controller until a break signal is outputted again.

4. The apparatus of claim 3, wherein the host computer recognizes a data flow and change by the address and the data outputted from the memory break controller.

5. The apparatus of claim 1, wherein the memory break controller comprises:

a memory break control register being activated by the break point address and the control signal inputted from the debugger controller;

an address register for storing the break point address inputted from the memory break control register;

an address comparator for comparing the address of the data memory currently used by the processor core and the break point address stored in the address register;

a data register for storing the data of the break point address stored in the address register; and

a data comparator for comparing the data of the current address outputted from the data memory and the data of the break point address stored in the data register.

6. The apparatus of claim 5, wherein the memory break control register comprises:

a memory break enable flag for activating the memory break controller;

a data check flag for sensing an address of the data memory which is identical to the break point address stored in the address register, and being enabled when the content of the data of the sensed address is outputted; and

an address trace check flag for being enabled when the content of the address of the data memory which is identical to the break point address stored in the address register is outputted.

7. The apparatus of claim 6, wherein when the address trace check flag is enabled, it outputs addresses and data of every memory which are read from or written in the processor core before the content of the break address is updated.

8. A debugging apparatus for informing a data transition state, comprising:

a debugger controller for outputting a control signal for performing a debugging, a memory break point address, and a break enable signal;

a processor core being activated by the control signal outputted from the debugger controller, and reading a data stored in a data memory or writing a data;

a memory break control register being activated by the break point address and the control signal inputted from the debugger controller;

an address register for storing the break point address inputted from the memory break control register;

an address comparator for comparing the address of the data memory

which is currently used by the processor and the break point address stored in the address register;

a data register for storing the data of the break point address stored in the address register; and

a data comparator for comparing the data of the current address outputted from the data memory and the data of the break point address stored in the data register.

9. A debugging method comprising the steps of:

outputting an address of a data memory to be observed, that is a break point address and a break enable signal, when a processor is switched to a debugging mode;

storing the outputted break point address, and operating the processor in a general operation state;

comparing the stored break point address and the address of the data memory currently used by the processor core, while the process is being operated;

outputting a break signal to suspend the process core, if the address of the data memory currently used by the processor core and the stored break point address are identical to each other; and

suspending the processor core by the outputted break signal and switching the processor to a debugging mode to debug the program.

10. The method of claim 9, wherein in the step of operating a processor, a memory break enable flag of a memory break controller is enabled according to an outputted break enable signal, and a data check flag and an address check flag are disabled, in order to initialize the processor, and then the break point address is stored.

11. The method of claim 9, wherein the step of outputting a break signal comprises:

a step in which when an address of the data memory currently used by the processor core and the stored break point address are identical to each other, it is determined whether the processor reads the data stored in the corresponding address or writes a data;

a step in which, in case of reading a data, the address trace check flag and the data check flag of the memory break control register are enabled;

a step in which the corresponding address of the data memory and the stored break point address are compared again; and

a step in which, if the corresponding address of the data memory and the break point address are not identical to each other, the corresponding address and data of the data memory are transmitted to the host computer.

12. The method of claim 11, wherein the step of comparing address comprises:

a step in which, if the corresponding address of the data memory and the

break point address are identical to each other, it is determined whether the processor reads a data stored in the data memory of the corresponding address or writes a data in the data memory of the corresponding address; and

a step in which, in case of writing a data, the data check flag of the memory break controller is enabled and a break signal for suspending the processor core is outputted.

13. The method of claim 12, wherein, in case of reading a data upon judgement, an address of the next data memory to be used by the processor core and the break point address are compared.

14. The method of claim 11, wherein, in case of writing a data in the judging step, the data check flag of the memory break controller is set and the operation of the processor core is discontinued.

15. The method of claim 9, wherein, in the step of outputting a break signal, when the operation of the processor core is suspended by the break signal, the address and data of the corresponding data memory are transmitted, and then the operation of the processor core is activated to output an address and a data of the data memory used by the processor core until a break signal is outputted again.

16. A debugging method for informing a data transition state, comprising:

a step in which when a processor is switched into a debugging mode, an address of a data memory to be observed, that is, a break point address, and a break enable signal are outputted;

a step in which the outputted break point address is stored;

a step in which the stored break point address and an address of a data memory currently used by a processor core are compared;

a step in which, when the address of the data memory currently used by the processor core and the stored break point address are identical to each other, it is determined whether the processor core reads a data stored in the corresponding address or writes a data;

a step in which, in case of writing a data in a corresponding address, a data check flag of a memory break controller is enabled and a break signal for suspending the processor core is outputted;

a step in which, when the processor core is suspended by the outputted break signal, the address and the data of the corresponding data memory are outputted; and

a step in which the operation of the processor core is activated and the address and the data of the data memory used by the processor are outputted until a break signal is outputted again.

17. The method of claim 16, wherein, in case of reading a data upon judgement, an address of the next data memory used by the processor core and the break point address are compared.

ABSTRACT OF THE DISCLOSURE

A debugging apparatus includes: a processor core operated by a program stored in a program memory to read a data stored in a data memory or write a data; a debugger controller for performing a debugging on the processor core upon receipt of a command from a host computer and outputting a data break point address; and a memory break controller for observing an address of a data memory used by the processor core, recognizing an address as a break point address to output a break signal to the debugger controller and the processor core to suspend the operation of the processor core, when the address is sensed to be identical, and transmitting a corresponding address and data to the host computer through the debugger controller. Since an address and a data of a specific data memory are monitored to recognize a data flow and change of the specific address, an error that an erroneous calculation is inputted a during processing or a data memory is erroneously assigned is quickly sensed. Thus, a time and an expense for a debugging operation can be much saved. In addition, by adding a data debugging method to the conventional program debugging method, a program development environment is set similar to an environment substantially operated by the processor. Thus, a time and an expense for developing a program can be also saved.